Claims

1. A client-server computing process wherein at least one server responds to requests from clients by returning information to clients, the computing process comprising:

initiating a request at a client;

communicating the request to the server;

responding to the request at the server by returning information to the client, wherein the information returned goes through at least parsing, layout, and rendering processes before being displayed at the client;

configuring the software carrying out at least one of the parsing and layout processes so that the location at which the process is performed can be changed between server and client at run time;

making a load-balancing determination as to whether the process should be run at the server or client; and

running the process at the chosen location.

2. The computing process of claim 1 wherein the client-server computing process is a web browsing process, and the server and clients are a browser server and a browser client.

3. The computing process of claim 2 wherein the browser server communicates with a web server to retrieve information.

4. The computing process of claim 1 wherein the load-balancing determination is based at least in part on a quality of service determination of the quality of service provided by one or both of the client and server.

5. The computing process of claim 4 wherein the quality of service determination is based on latency of processes carried out on one or both of the client and server.

6. The computing process of claim 1 wherein the load-balancing determination is based at least in part on the load of one or both of the client and server.

7. The computing process of claim 1 wherein the load-balancing determination is based at least in part on the configuration of the clients.

8. The computing process of claim 7 wherein the load-balancing determination is further based at least in part on the configuration of the server.

9. The computing process of claim 7 or 8 wherein the configuration on which the load-balancing determination is based changes dynamically and the load-balancing determination responds dynamically to such changes.

10. The computing process of claim 7 or 8 wherein the configuration on which the load-balancing determination is based is assumed to remain static after a load-balancing determination based on such configuration is made.

11. The computing process of claim 7 wherein the load-balancing determination is based only on the configuration of the clients, and remains fixed during operation.

12. The computing process of claim 7 wherein the load-balancing determination is based on both the configuration of the clients and a quality of service determination of the quality of service provided by one or both of the client and server.

13. The computing process of claim 12 wherein the quality of service determination is based on latency of processes carried out on one or both of the client and server.

14. The computing process of claim 7 wherein the load-balancing determination is based on both the configuration of the clients and on the load of one or both of the client and server.

15. The computing process of claim 1 wherein both the layout and parsing processes are configured so that the location at which both processes are run may be changed between server and clients.

16. The computing process of claim 1 wherein clients with different configurations use the same server so that an initial load balancing determination based on those configuration differences locates processes between client and server differently for differently configured clients.

17. The computing process of claim 16 wherein during operation further changes are made to the location at which processes are carried out based on quality of service determinations.

18. The computing process of claim 5 wherein latency is measured using a timecard.

19. The computing process of claim 1 wherein the processes that are carried out at either server or client comprise distributed objects that migrate between client and server.

20. The computing process of claim 19 wherein the objects are Java Beans processed in containers.

21. The computing process of claim 1 further comprising a fetching process that can be run at either the client or the server based on the outcome of a load balancing determination.

22. The computing process of claim 1 wherein the rendering process is always performed at the client.

23. The computing process of claim 1 further comprising a script evaluation and execution process that can be run at either the client or the server based on the outcome of a load balancing determination.

24. The computing process of claim 1 wherein information is cached at the client, and the type of information cached varied depending on which processes are running the client.

25. The computing process of claim 1 wherein the load balancing determination is based on one or more of the following: client computational resources, client load, server computational resources, server load, number of clients per server, network traffic between clients and server, and security.

26. The computing process of claim 1 wherein the load balancing determination is based on one or more of the following: latency of processing a request downstream/upstream from a given process, and latency of processing a request of a given process.

27. The computing process of claim 1 wherein the processes are pre-configured on the clients and server, so that they may be run on demand, by activating a process on one of the client and server, and deactivating the corresponding process on the other of the client and server, approximately simultaneously.

28. The computing process of claim 1 wherein the ability to change the location at which processes are run may be locked to maintain the distribution of processes in a selected distribution.

29. The computing process of claim 1 wherein the processes that run at the server are interconnected by a switch.